This listing of claims replaces all prior versions, and listings of claims in the instant application:

Listing of Claims:

1.   (Currently Amended)   A method for processing a document page containing transparent graphics objects for printing, said method comprising:

creating a transparency list containing a reference to each of the transparent graphics objects of said document page,

wherein a transparent graphics object is an object, which does not obscure any other object that overlaps the object; and

determining whether said transparency list is empty;

converting only objects contained in referenced in said transparency list into bitmaps for printing upon said determining finding said transparency list is not empty; and

printing said document page, without performing said converting upon said determining finding said transparency list is empty.

2.   (Original)   The method of Claim 1, further comprising:

generating a bitmap frame for at least one object in said transparency list wherein an area of said bitmap frame is less than an area of the entire document page.

3.   (Original)   The method of Claim 1, further comprising:

processing portions of the document page that do not include objects in said transparency list as non-bitmapped data.

4. (Currently Amended) The method of Claim 1, further comprising:

examining a non-transparent object of said document page for overlap with at least one transparent graphics object ~~contained~~ referenced in said transparency list; and

inserting said non-transparent object into said transparency list upon said non-transparent object overlapping with at least one transparent graphics object ~~contained~~ referenced in said transparency list.

5. (Original) The method of Claim 1, further comprising:
generating a frame for each overlapping compound object in said transparency list, an overlapping compound object being formed by a transparent graphics object and non-transparent objects overlapping said transparent graphics object, said frame defining an area to be printed as a bitmap in printing the overlapping compound object.

6. (Original) The method of Claim 5, further comprising:
generating said frame for each overlapping compound object as a composition of subframes.

7. (Currently Amended) A system comprising:
a processor; and
a memory coupled to said processor, and storing therein computer instructions for a method for processing a document page containing transparent graphics objects for printing wherein upon execution of said computer instructions on said processor, said method comprises:
creating a transparency list containing a reference to each of the transparent graphics objects of said document page in said memory,

wherein a transparent graphics object is an object, which does not obscure any other object that overlaps the object; ~~and~~
determining whether said transparency list is empty;
converting ~~only~~ objects ~~contained in~~ referenced in said transparency list into bitmaps for printing upon said ~~determining~~ finding said transparency list is not empty; and
printing said document page, without performing said converting upon said ~~determining~~ finding said transparency list is empty.

8. (Original) The system of Claim 7, wherein said method further comprises:
generating a bitmap frame for at least one object in said transparency list wherein an area of said bitmap frame is less than an area of the entire document page.

9. (Original) The system of Claim 7, wherein said method further comprises:
processing portions of the document page that do not include objects in said transparency list as non-bitmapped data.

10. (Currently Amended) The system of Claim 7, wherein said method further comprises:
examining a non-transparent object of said document page for overlap with at least one transparent graphics object ~~contained~~ referenced in said transparency list; and
inserting said non-transparent object into said transparency list upon said non-transparent object

overlapping with at least one transparent graphics object ~~contained~~ <u>referenced</u> in said transparency list.

11.   (Original)   The system of Claim 7, wherein said method further comprises:

generating a frame for each overlapping compound object in said transparency list, an overlapping compound object being formed by a transparent graphics object and non-transparent objects overlapping said transparent graphics object, said frame defining the area to be printed as a bitmap for printing the overlapping compound object.

12.   (Original)   The system of Claim 11, wherein said method further comprises:

generating said frame for each overlapping compound object as a composition of subframes.

13.   (Currently Amended)   A computer program product comprising computer program code for a method for processing a document page containing transparent graphics objects for printing, said method comprising:

creating a transparency list containing <u>a reference to each of</u> the transparent graphics objects of said document page<u>,</u>

<u>wherein a transparent graphics object is an object, which does not obscure any other object that overlaps the object;</u> ~~and~~

<u>determining whether said transparency list is empty;</u>

converting ~~only~~ objects ~~contained in~~ <u>referenced in</u> said transparency list into bitmaps for printing <u>upon said determining</u> ~~finding~~ said transparency list is not empty<u>;</u>

<u>and</u>

printing said document page, without performing said converting upon said determining finding said transparency list is empty.

14. (Original) The computer-program product of Claim 13, wherein said method further comprises:

generating a bitmap frame for at least one object in said transparency list wherein an area of said bitmap frame is less than an area of the entire document page.

15. (Original) The computer-program product of Claim 13, wherein said method further comprises:

processing portions of the page that do not include objects in said transparency list as non-bitmapped data.

16. (Currently Amended) The computer-program product of Claim 13, wherein said method further comprises:

examining a non-transparent object of said document page for overlap with at least one transparent graphics object ~~contained~~ referenced in said transparency list; and

inserting said non-transparent object into said transparency list upon said non-transparent object overlapping with at least one transparent graphics object ~~contained~~ referenced in said transparency list.

17. (Original) The computer-program product of Claim 13, wherein said method further comprises:

generating a frame for each overlapping compound object in said transparency list, an overlapping compound object being formed by a transparent graphics object and non-transparent objects overlapping said transparent graphics object, said frame defining the area to be

printed as a bitmap for printing the overlapping compound object.

18.   (Original)   The computer-program product of Claim 17, wherein said method further comprises:
    generating said frame for each overlapping compound object as a composition of subframes.

19.   (Currently Amended)   A method for processing a document page containing transparent graphics objects for printing, said method comprising:
    determining whether at least a part of a non-transparent object including a plurality of text words overlaps a transparent graphics object ~~contained~~ referenced in a transparency list wherein said transparent graphics object is an object, which does not obscure any other object that overlaps the object; and
    adding said entire non-transparent object to said transparency list when either of (a) all of said plurality of text words of said non- transparent object overlaps said transparent graphics object; and (b) only said at least a part of said plurality of text words of said non-transparent object overlaps said transparent graphics object is true.

20.   (Currently Amended)   The method of Claim 19 further comprising:
    creating said transparency list ~~containing said transparent graphics object prior to said determining operation~~.